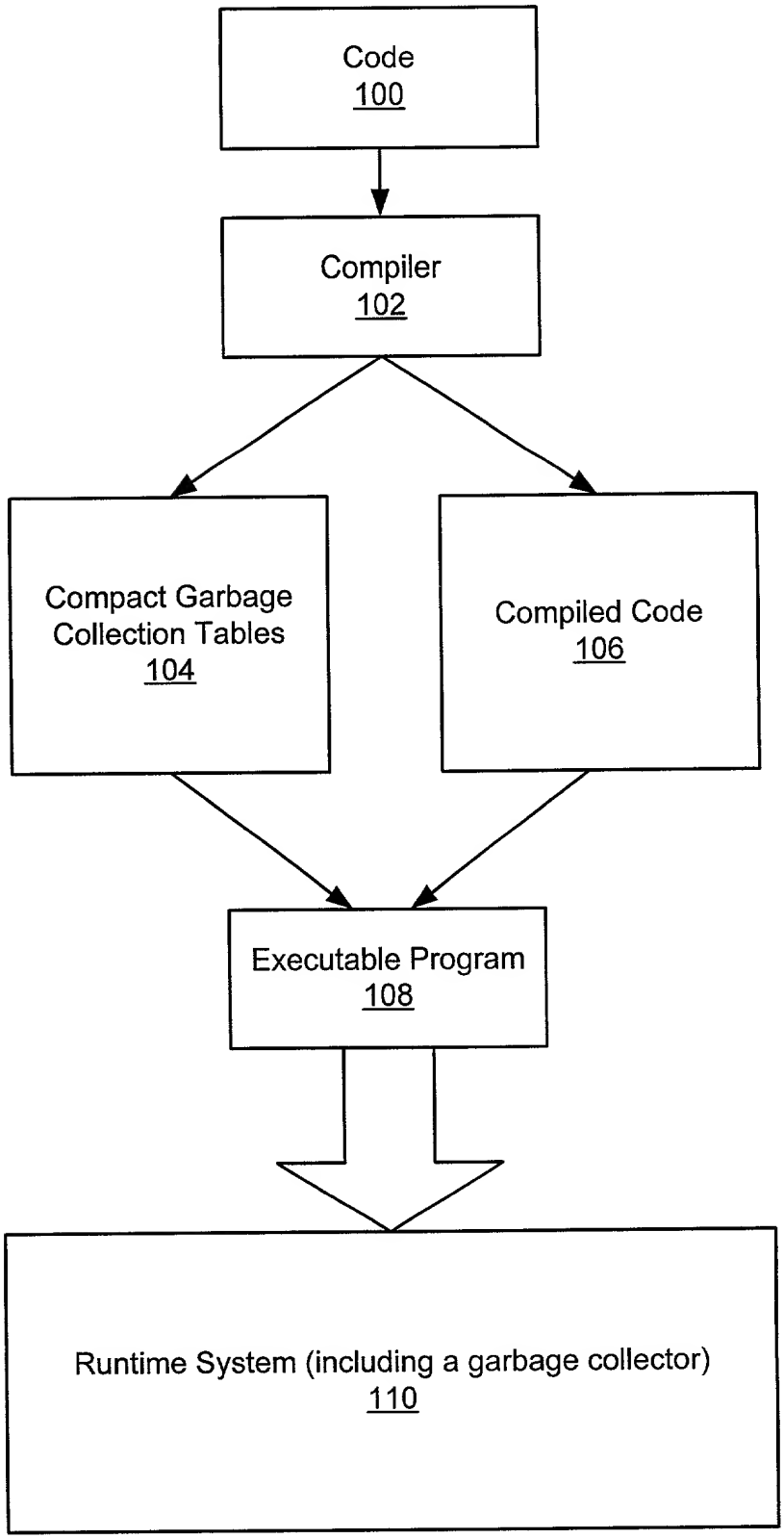
Code
100
Compiler
102
Compact Garbage Collection Tables
104
Compiled Code
106
Executable Program
108
Runtime System (including a garbage collector)
110

FIG. 1

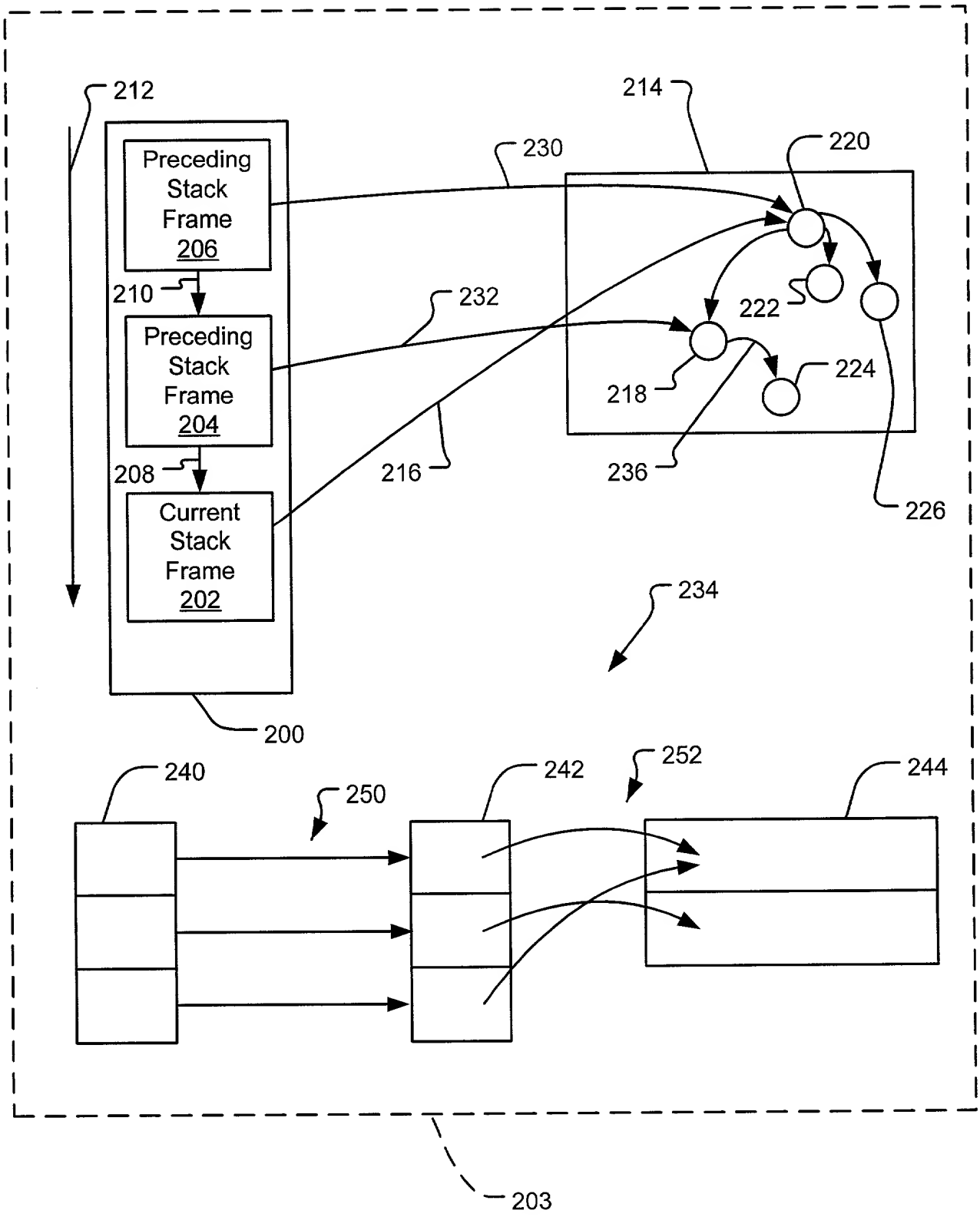
212
214
Preceding Stack Frame 206
230
220
210
232
222
Preceding Stack Frame 204
218
224
208
216
236
226
Current Stack Frame 202
234
200
240
250
242
252
244
203

FIG. 2

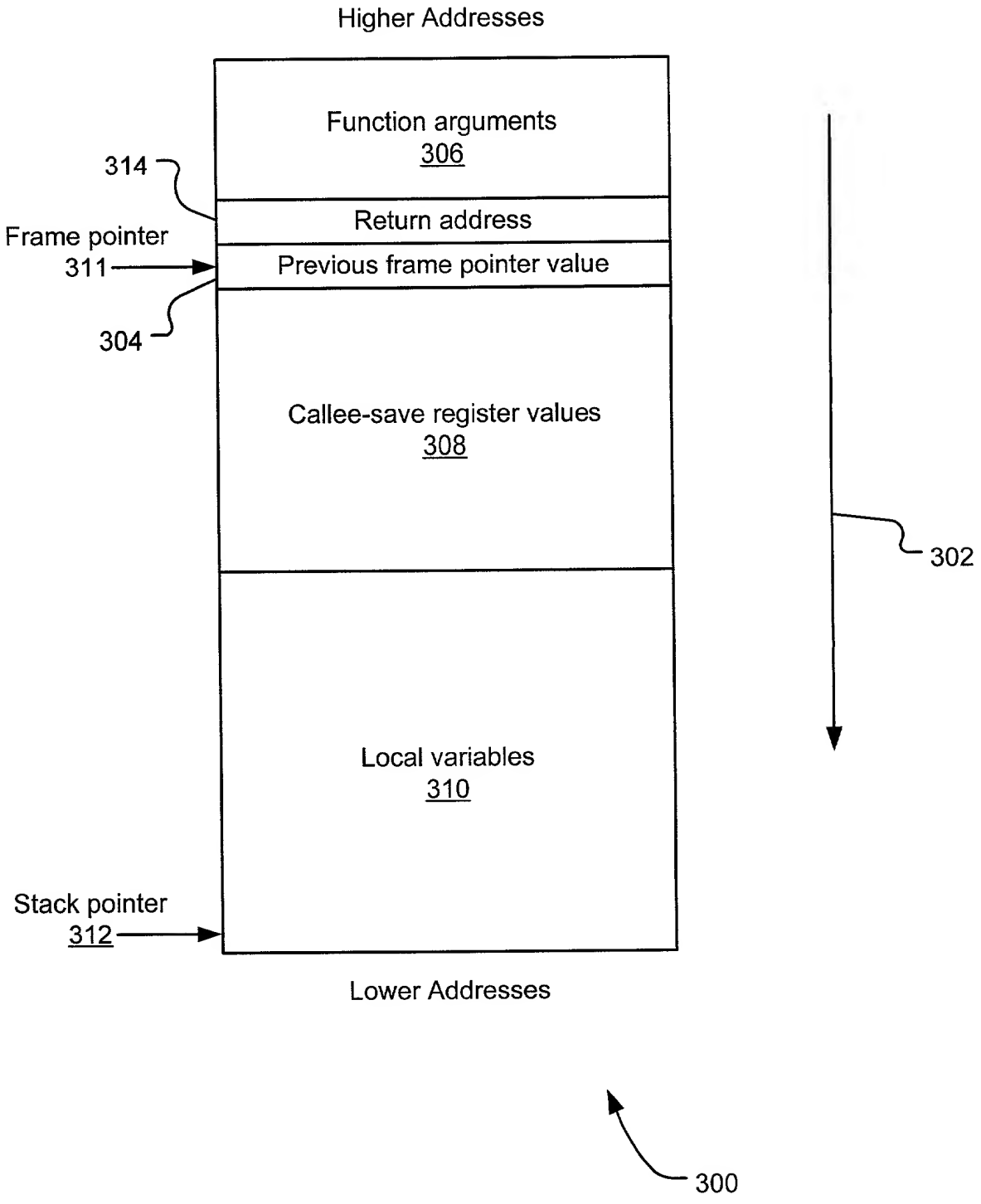
Higher Addresses
Function arguments
306
314
Return address
Frame pointer
311
Previous frame pointer value
304
Callee-save register values
308
302
Local variables
310
Stack pointer
312
Lower Addresses
300

FIG. 3

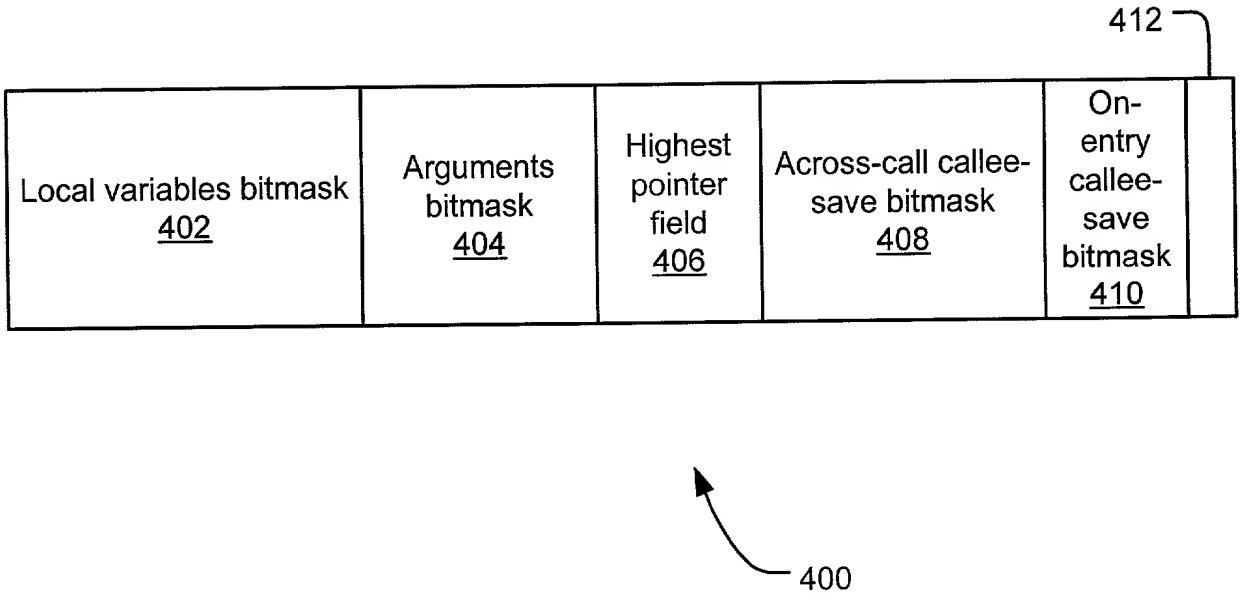
412
Local variables bitmask
402
Arguments bitmask
404
Highest pointer field
406
Across-call callee-save bitmask
408
On-entry callee-save bitmask
410
400

FIG. 4

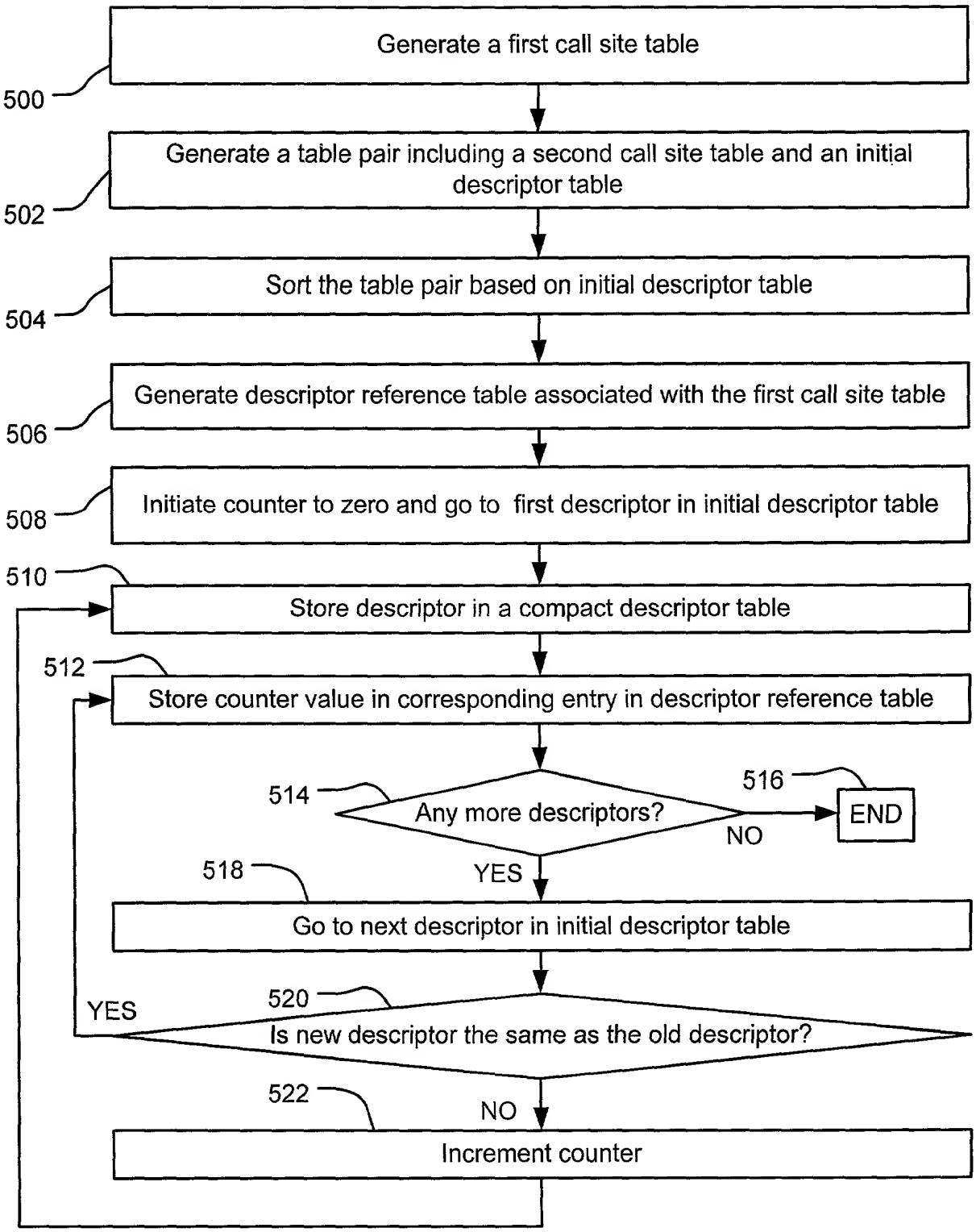
Generate a first call site table
500
Generate a table pair including a second call site table and an initial descriptor table
502
Sort the table pair based on initial descriptor table
504
Generate descriptor reference table associated with the first call site table
506
Initiate counter to zero and go to first descriptor in initial descriptor table
508
510
Store descriptor in a compact descriptor table
512
Store counter value in corresponding entry in descriptor reference table
514
Any more descriptors?
516
END
NO
YES
518
Go to next descriptor in initial descriptor table
520
YES
Is new descriptor the same as the old descriptor?
522
NO
Increment counter

FIG. 5

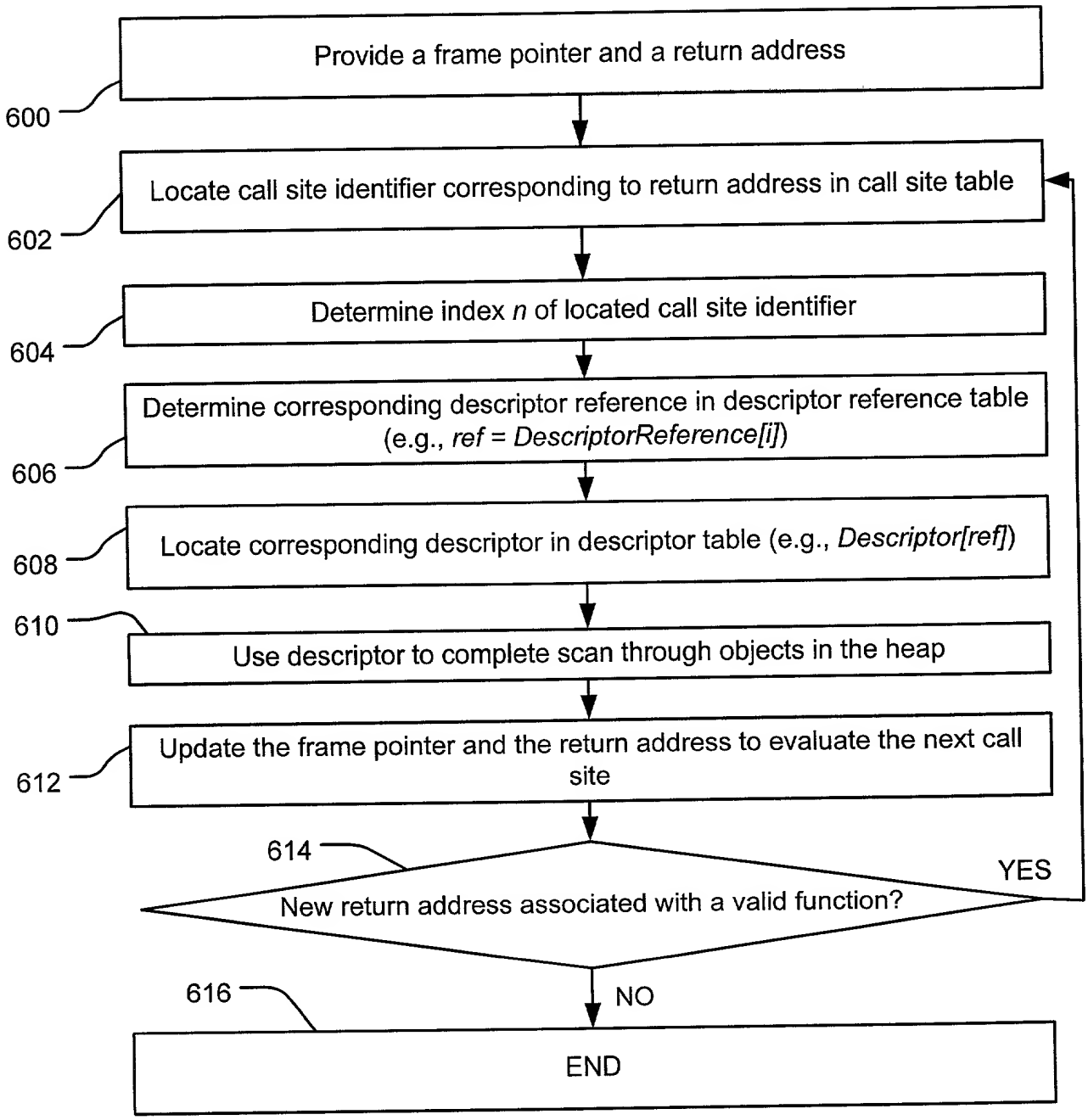
Provide a frame pointer and a return address
600
Locate call site identifier corresponding to return address in call site table
602
Determine index n of located call site identifier
604
Determine corresponding descriptor reference in descriptor reference table (e.g., ref = DescriptorReference[i])
606
Locate corresponding descriptor in descriptor table (e.g., Descriptor[ref])
608
610
Use descriptor to complete scan through objects in the heap
Update the frame pointer and the return address to evaluate the next call site
612
614
New return address associated with a valid function?
YES
NO
616
END

FIG. 6

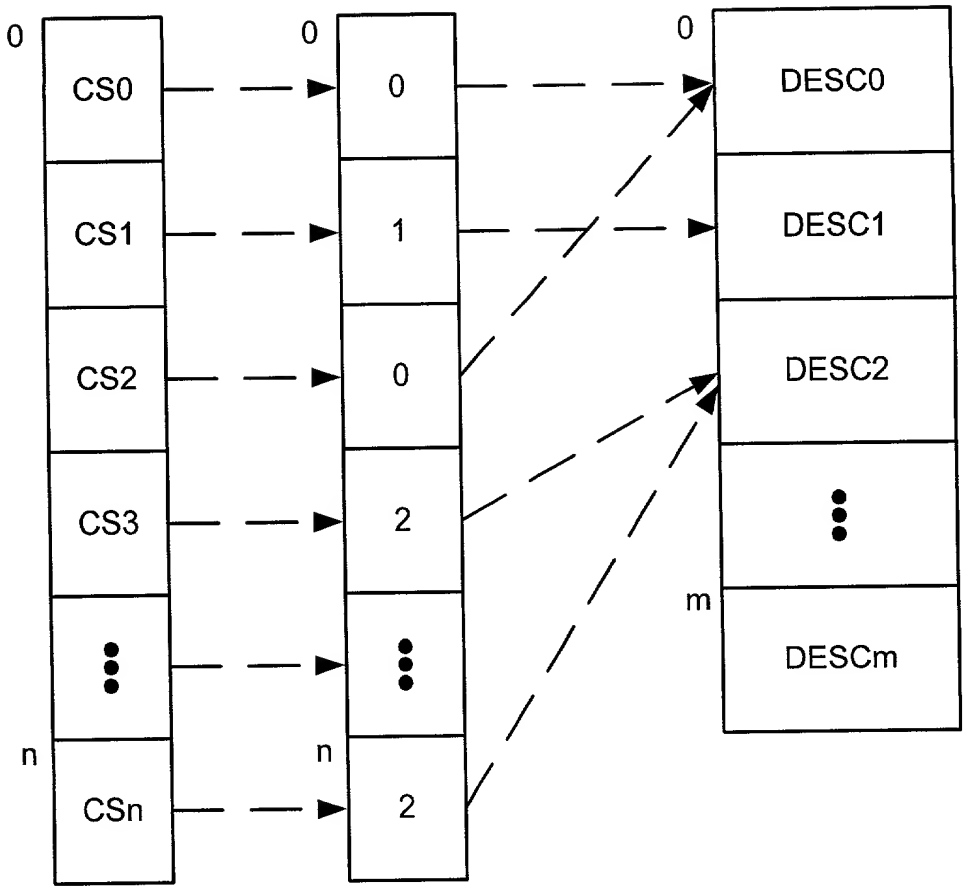

FIG. 7

TOOE20"98T6T660
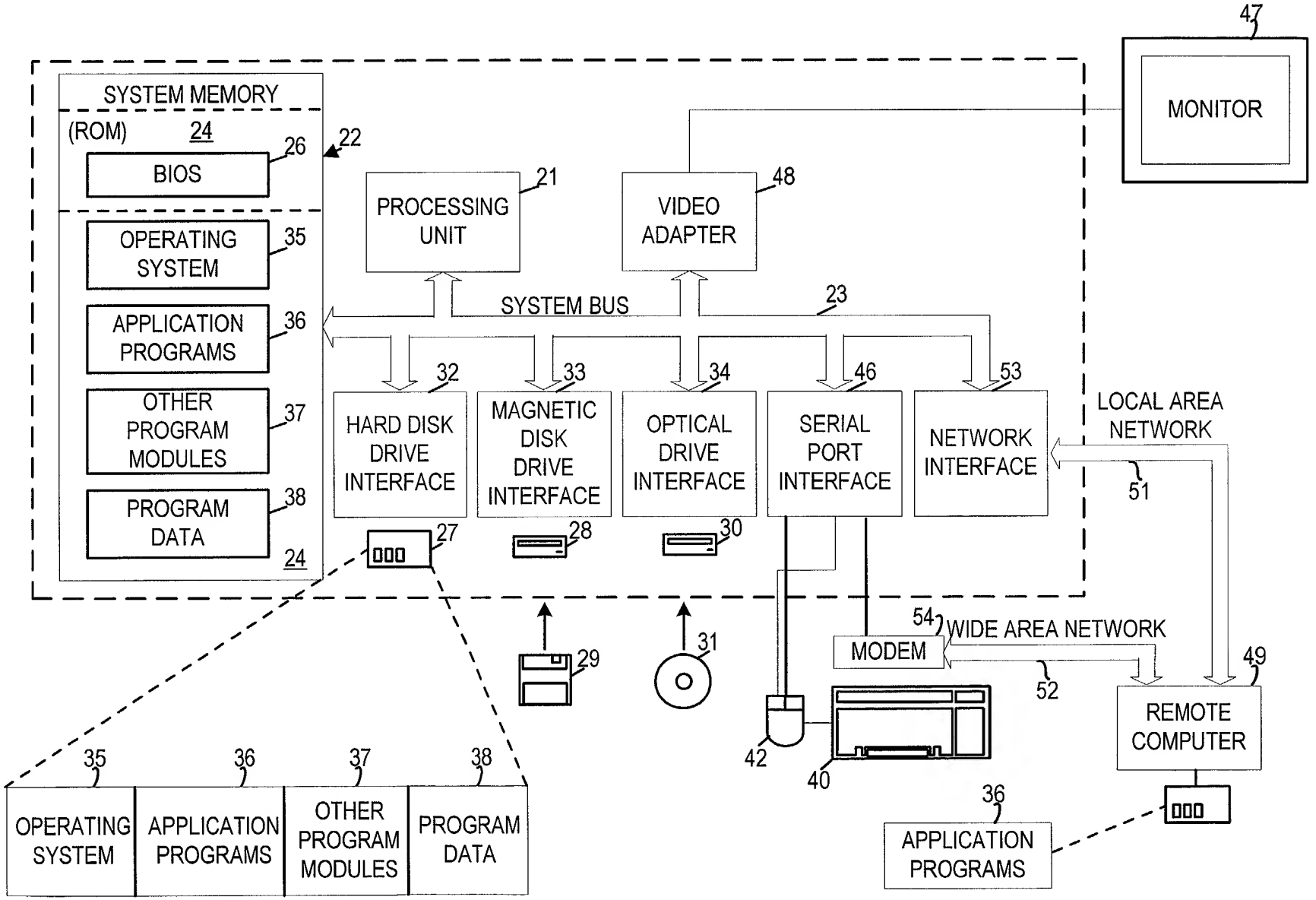
FIG. 8